CMU OCI Challenge 2016

00:59:32
to test end

Maximum Difference in an Array

Given an array of integers, ***a***, return the **maximum difference** of any pair of numbers **such that the larger integer in the pair occurs at a higher index (in the array) than the smaller integer.** Return -1 if you cannot find a pair that satisfies this condition.

You are provided with a function **maxDifference** which takes in the array as a parameter. You are required to return an integer which is the right answer. The code for parsing the input and displaying the output is provided, so your only task is to complete the body of the provided function so it returns the correct output to the caller. However, for completeness, we describe the input format:

Constraints:

$1 \leq N \leq 1{,}000{,}000$

$-1{,}000{,}000 \leq a[i] \leq 1{,}000{,}000 \ \forall\, i \in [0, N-1]$

Input Format:

The first line of the input is N (the number of elements in the array), and then followed by N elements each in a separate line.

Sample Input 0:

```
7
2
3
10
2
4
8
1
```

Sample Output 0:

```
8
```

Explanation 0:

For the array { 2,3,10,2,4,8,1} given above, *10* is the largest number in the array and *1* is the smallest number in the array. However, the index of *10* is lower than the lowest index that contains a *1* so the condition of the problem is not satisfied. Using zero-based index notation, the correct answer is ***a**[2] - **a**[0] = 10 - 2 = 8*. This satisfies the condition that the larger number in the pair should be positioned at a higher index in the array than the smaller number.

Sample Input 1:

6
7
9
5
6
3
2

Sample Output 1:

2

Explanation 1:

The **value of maxDifference** is *9 - 7 = 2*.
9 occurs at ***a**[1]* and *7* occurs at ***a**[0]*. This satisfies the condition that the larger number must have a higher index than the smaller number.

YOUR ANSWER

Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

/*
 * Complete the function below.
 */

    static int maxDifference(int[] a) {


    }
```

```
19      public static void main(String[] args) throws
    IOException{
20          Scanner in = new Scanner(System.in);
21          final String fileName =
    System.getenv("OUTPUT_PATH");
22          BufferedWriter bw = new BufferedWriter(new
    FileWriter(fileName));
23          int res;
24
25          int _a_size = 0;
26          _a_size = Integer.parseInt(in.nextLine());
27          int[] _a = new int[_a_size];
28          int _a_item;
29          for(int _a_i = 0; _a_i < _a_size; _a_i++) {
30              _a_item = Integer.parseInt(in.nextLine());
31              _a[_a_i] = _a_item;
32          }
33
34          res = maxDifference(_a);
35          bw.write(String.valueOf(res));
36          bw.newLine();
37
38          bw.close();
39      }
40  }
```

Line: 12 Col: 1

Test against custom input

Run Code

Submit code & Continue

Download sample testcases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*